

## Задача А. Pink Floyd

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

*Это интерактивная задача.*

Учёные находятся всего в нескольких шагах от открытия новой оптимизации алгоритма Флойда, позволяющей ему работать за линейное время! Осталась лишь одна часть.

Как известно, для запуска алгоритма Флойда в графе на  $n$  вершинах должно быть по одному ребру между любыми двумя вершинами. У ученых есть такой граф, однако, они ориентировали каждое ребро, то есть направили его ровно в одну из двух сторон.

Для оптимизации Флойда выбраны  $m$  рёбер и покрашены в розовый цвет, остальные ребра покрашены в зеленый. Вы знаете направление каждого из этих  $m$  ребер, однако, направление зеленых рёбер вам заранее неизвестно. За один запрос вы можете узнать направление одного зеленого ребра, но всего вы можете сделать не более  $2 \cdot n$  запросов.

Вам нужно найти такую вершину, от которой любая другая вершина достижима по пути, состоящем из ребер одного цвета. Обратите внимание, ученые могут лукавить, что уже определили направления для всех ребер, и выдавать ответы в зависимости от ваших запросов.

### Формат входных данных

В первой строке находятся два целых числа  $n$  и  $m$  ( $1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 100\,000$ ) — число вершин и число рёбер розового цвета.

В  $i$ -й из следующих  $m$  строках находятся два целых числа  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ) — начало  $i$ -го ребра розового цвета и его конец. Гарантируется, что все неупорядоченные пары  $(u_i, v_i)$  различны.

### Формат выходных данных

Когда вы найдете ответ, выведите «!» и номер вершины, от которой каждая другая достижима по одноцветному пути.

### Протокол взаимодействия

Чтобы узнать направление зеленого ребра между вершинами  $a$  и  $b$ , в одной строке через пробел выведите символ «?», а затем два целых числа  $a$  и  $b$ .

В ответ на это считайте целое число, которое будет равно 1, если ребро направленно от  $a$  к  $b$ , и 0, если ребро направлено от  $b$  к  $a$ .

Вы можете задать не более  $2 \cdot n$  вопросов, в противном случае вы получите вердикт **Wrong Answer**.

После вывода запроса не забудьте вывести перевод строки и сбросить буфер вывода. В противном случае вы получите вердикт **Решение «зависло»**. Для сброса буфера используйте:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `stdout.flush()` в Python;
- смотрите документацию для других языков.

Ответ  $-1$  вместо 0 или 1 означает, что ваша программа сделала некорректный запрос или превысила число запросов. Ваша программа должна немедленно завершиться после прочтения ответа  $-1$ , вы получите вердикт **Неправильный ответ**. В противном случае вы можете получить любой вердикт, так как программа продолжит чтение из закрытого потока.

### Взломы

Для взлома задайте тест в следующем формате. В первой строке должны находиться два целых числа  $n$  и  $m$  ( $1 \leq n \leq 300$ ,  $0 \leq m \leq n \cdot (n - 1) / 2$ ) — число вершин и число рёбер розового цвета.

В  $i$ -й из следующих  $m$  строк должны находиться два целых числа  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i$ ), означающих, что есть ребро розового цвета от вершины  $a_i$  к вершине  $b_i$ . Все неориентированные пары  $(a_i, b_i)$  должны быть различны.

В  $i$ -й из следующих  $(n \cdot (n - 1)/2 - m)$  строках должны находиться два целых числа  $a_i, b_i$  ( $1 \leq a_i, b_i \leq n, a_i \neq b_i$ ), означающих, что есть ребро зеленого цвета от вершины  $a_i$  к вершине  $b_i$ . Все неориентированные пары  $(a_i, b_i)$  должны быть различны и не должны совпадать с теми же парами для рёбер розового цвета.

### Пример

стандартный ввод	стандартный вывод
4 2	? 1 3
1 2	? 4 2
3 4	? 3 2
0	! 3
1	
1	

### Замечание

В приведённом примере ответ на запрос «? 1 3» — 0, поэтому ребро ориентированно от 3 к 1; ответ на запрос «? 4 2» — 1, поэтому ребро ориентированно от 4 к 2; ответ на запрос «? 3 2» — 1, поэтому ребро ориентированно от 3 к 2. Тогда от вершины 3 есть пути по зелёным рёбрам к вершинам 1 и 2, а так же есть путь по розовым рёбрам до вершины 4.

## Задача В. Наркоконтроль

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В маленьком городке «М» начала действовать служба наркоконтроля. Первая задача службы — выяснить, сколько наркоторговцев работает в окрестности города. Агенты службы опросили всех наркозависимых в городе и составили список случаев продажи травки, произошедших за одни сутки, с указанием места и времени наблюдения. Теперь аналитики хотят понять, сколько же на самом деле есть наркоторговцев. Из данных разведки известна максимальная скорость, с которой может двигаться наркоторговец. Аналитики просят вас узнать, какое минимальное количество наркоторговцев могли участвовать во всех зафиксированных случаях продажи травки.

### Формат входных данных

На первой строке входного файла содержатся целые числа  $n$  и  $v$  — количество случаев продажи травки и максимальная скорость наркоторговца ( $1 \leq n \leq 100, 1 \leq v \leq 10000$ ). Следующие  $n$  строк содержат описания случаев продажи травки в формате «ЧЧ:ММ  $x$   $y$ », где ЧЧ:ММ — время продажи,  $x$  и  $y$  — координаты места, в котором продавалась травка (для простоты будем считать, что всё происходило на плоскости). Координаты по модулю не превышают 1000. Скорость выражена в км/ч, координаты — в км.

### Формат выходных данных

Выведите в выходной файл одно число — минимальное возможное количество наркоторговцев.

### Пример

стандартный ввод	стандартный вывод
4 1 12:00 0 0 13:10 0 1 14:00 1 0 15:00 1 1	2

## Задача С. Экспедиция

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Винни-Пух отправился в очередную Экспедицию по Лесу!

Лес представляет собой  $n$  полянок, соединённых  $m$  двусторонними тропинками. При этом возможно, что для некоторых полянок не существует способа добраться из одной в другую по тропинкам. За половину суток (день или ночь) Пух проходит ровно по одной тропинке.

Когда Пух пришёл на рассвете на очередную полянку, ему показалось, что он вернулся к полянке, с которой он начал свое путешествие. Он также уверен, что не проходил ни по одной тропинке дважды, и что все полянки до этого были различными. Также он помнит, что он отправился в Экспедицию на рассвете, то есть число дней в пути было равно числу ночей.

Вы решили помочь медвежонку и хотите определить по карте Леса, могло ли такое случиться.

### Формат входных данных

Первая строка содержит число  $T$  — число наборов входных данных, которые вам предстоит обработать. Далее следуют  $T$  наборов входных данных. Каждый набор задаётся в следующем формате.

В первой строке набора входных данных заданы два числа  $n$  и  $m$  ( $2 \leq n \leq 500\,000$ ,  $1 \leq m \leq 500\,000$ ) — количество полянок и количество тропинок в соответствующем Лесу.

В следующих  $m$  строках содержатся описания тропинок: каждая строка состоит из двух целых чисел  $u_i, v_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ), обозначающих номера полянок, соединённых очередной тропинкой.

Гарантируется, что никакие две полянки не соединены двумя или более тропинками.

Гарантируется, что и суммарное число полянок, и суммарное число тропинок по всем наборам входных данных не превзойдёт 500 000.

### Формат выходных данных

Для каждого набора входных данных выведите в отдельной строке «YES», если в Лесу существует маршрут, удовлетворяющий описанию Пуха, и «NO» в противном случае.

### Пример

стандартный ввод	стандартный вывод
2	YES
5 6	NO
1 2	
2 3	
3 4	
4 5	
5 2	
5 3	
3 3	
1 2	
2 3	
3 1	

### Замечание

Тест из условия состоит из двух наборов входных данных.

В первом примере Пух мог начать Экспедицию на полянке с номером 2, в первый день перейти на полянку 5, в первую ночь перейти на полянку 3, во второй день перейти на полянку 4 и во вторую ночь вернуться обратно на полянку 2.

Во втором примере, независимо от выбора начальной полянки, Пух может вернуться на неё, но ему в любом случае потребуется три перехода, а значит дней в пути будет больше, чем ночей.

## Задача D. Xor the Graph

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дан неориентированный граф, состоящий из  $N$  вершин и  $M$  рёбер. Каждому ребру присвоено значение, 0 или 1. Операция состоит в том, чтобы выбрать путь (не обязательно простой), и изменить значения, записанные на каждом его ребре на противоположные. Сделайте значения на всех рёбрах нулями за минимальное число операций.

### Формат входных данных

На первой строке заданы два целых числа  $N$  и  $M$  ( $1 \leq N \leq 10^5$ ,  $0 \leq M \leq 10^5$ ).

В каждой из следующих  $M$  строк записаны по три целых числа  $a$ ,  $b$  и  $c$ . Они означают, что существует ребро между вершинами  $(a, b)$ , изначальное значение которого равно  $c$  ( $1 \leq a, b \leq n$ ,  $0 \leq c \leq 1$ ).

### Формат выходных данных

Выведите количество операций  $K$  на первой строке.

В последующих  $K$  строках выведите пути, которые вы выбрали:

- Первым числом выведите  $P$  — количество вершин в пути
- Затем выведите  $P$  целых чисел  $v_1, v_2, \dots, v_P$  — индексы вершин. Для всех  $1 \leq i \leq P - 1$   $(v_i, v_{i+1})$  должно являться ребром в заданном графе.

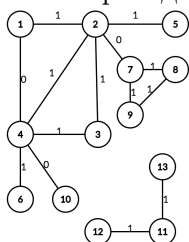
Решение будет считаться верным, если оно использует минимальное количество операций, и при этом  $\sum P \leq 4 * M$ .

### Пример

стандартный ввод	стандартный вывод
13 14	3
1 2 1	5 1 2 3 4 6
2 3 1	8 4 2 7 8 9 7 2 5
3 4 1	3 12 11 13
2 4 1	
1 4 0	
4 6 1	
4 10 0	
2 5 1	
2 7 0	
7 8 1	
8 9 1	
9 7 1	
11 12 1	
11 13 1	

### Замечание

Ниже приведён граф в первом тесте.



## Задача Е. День рождения

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Митя знаком с  $m$  юношами и  $n$  девушками и хочет пригласить часть из них на свой день рождения. Ему известно, с какими девушками знаком каждый юноша, и с какими юношами знакома каждая девушка. Он хочет добиться того, чтобы каждый приглашённый был знаком со всеми приглашёнными противоположного пола, пригласив при этом максимально возможное число своих знакомых. Помогите ему это сделать!

### Формат входных данных

Входной файл состоит из одного или нескольких наборов входных данных. В первой строке входного файла записано число наборов  $k$  ( $1 \leq k \leq 20$ ). В последующих строках записаны сами наборы входных данных.

В первой строке каждого набора задаются числа  $0 \leq m \leq 150$  и  $0 \leq n \leq 150$ . Далее следуют  $m$  строк, в каждой из которых записано одно или несколько чисел — номера девушек, с которыми знаком  $i$ -й юноша (каждый номер встречается не более одного раза). Строка завершается числом 0.

### Формат выходных данных

Для каждого набора выведите четыре строки. В первой из них выведите максимальное число знакомых, которых сможет пригласить Митя. В следующей строке выведите количество юношей и количество девушек в максимальном наборе знакомых. Следующие две строки должны содержать номера приглашённых юношей и приглашённых девушек соответственно. Если максимальных наборов несколько, то выведите любой из них.

### Пример

стандартный ввод	стандартный вывод
2	4
2 2	2 2
1 2 0	1 2
1 2 0	1 2
3 2	4
1 2 0	2 2
2 0	1 3
1 2 0	1 2

## Задача F. Цветные волшебники

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Сказочная страна представляет собой множество городов, соединенных дорогами с двухсторонним движением. Причем из любого города страны можно добраться в любой другой город либо непосредственно, либо через другие города. Известно, что в сказочной стране не существует дорог, соединяющих город сам с собой и между любыми двумя разными городами, существует не более одной дороги.

В сказочной стране живут желтый и синий волшебники. Желтый волшебник, пройдя по дороге, перекрашивает ее в желтый цвет, синий — в синий. Как известно, при наложении желтой краски на синюю, либо синей краски на желтую, краски смешиваются и превращаются в краску зеленого цвета, который является самым нелюбимым цветом обоих волшебников.

В этом году в столице страны (городе  $f$ ) проводится конференция волшебников. Поэтому желтый и синий волшебники хотят узнать, какое минимальное количество дорог им придется перекрасить в зеленый цвет, чтобы добраться в столицу. Изначально все дороги не покрашены.

Начальное положение желтого и синего волшебников заранее не известно. Поэтому необходимо решить данную задачу для  $k$  возможных случаев их начальных расположений.

### Формат входных данных

Первая строка входного файла содержит целые числа:  $n$  ( $1 \leq n \leq 10^5$ ) и  $m$  ( $1 \leq m \leq 5 \cdot 10^5$ ) — количество городов и дорог в волшебной стране соответственно.

Вторая строка содержит одно целое число  $f$  ( $1 \leq f \leq n$ ) — номер города, являющегося столицей сказочной страны.

В следующих  $m$  строках, находится описание дорог страны. В этих  $m$  строк записано по два целых числа  $a_i$  и  $b_i$ , означающих, что существует дорога, соединяющая города  $a_i$  и  $b_i$ .

Следующая строка содержит целое число  $k$  ( $1 \leq k \leq 10^5$ ) — количество возможных начальных расположений волшебников.

Далее следуют  $k$  строк, каждая из которых содержит два целых числа — номера городов, в которых изначально находится желтый и синий волшебники соответственно.

### Формат выходных данных

Для каждого из  $k$  случаев, ваша программа должна вывести в выходной минимальное количество дорог, которое придется покрасить в зеленый цвет волшебникам для того, чтобы добраться в столицу

### Пример

стандартный ввод	стандартный вывод
6 6	1
1	2
1 2	
2 3	
3 4	
4 2	
4 5	
3 6	
2	
5 6	
6 6	

## Задача G. Обновление дата-центров

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

У компании BigData Inc. есть  $n$  дата-центров, пронумерованных от 1 до  $n$ , расположенных по всему миру. В этих дата-центрах хранятся данные клиентов компании (как можно догадаться из названия — большие данные!)

Основой предлагаемых компанией BigData Inc. услуг является гарантия возможности работы с пользовательскими данными даже при условии выхода какого-либо из дата-центров компании из доступности. Подобная гарантия достигается путём использования *двойной репликации* данных. Двойная репликация — это подход, при котором любые данные хранятся в двух идентичных копиях в двух различных дата-центрах.

Про каждого из  $m$  клиентов компании известны номера двух различных дата-центров  $c_{i,1}$  и  $c_{i,2}$ , в которых хранятся его данные.

Для поддержания работоспособности дата-центра и безопасности данных программное обеспечение каждого дата-центра требует регулярного обновления. Релизный цикл в компании BigData Inc. составляет один день, то есть новая версия программного обеспечения выкладывается на каждый компьютер дата-центра каждый день.

Обновление дата-центра, состоящего из множества компьютеров, является сложной и длительной задачей, поэтому для каждого дата-центра выделен временной интервал длиной в час, в течение которого компьютеры дата-центра обновляются и, как следствие, могут быть недоступны. Будем считать, что в сутках  $h$  часов. Таким образом, для каждого дата-центра зафиксировано целое число  $u_j$  ( $0 \leq u_j \leq h-1$ ), обозначающее номер часа в сутках, в течение которого  $j$ -й дата-центр недоступен в связи с плановым обновлением.

Из всего вышесказанного следует, что для любого клиента должны выполняться условия  $u_{c_{i,1}} \neq u_{c_{i,2}}$ , так как иначе во время одновременного обновления обоих дата-центров, компания будет не в состоянии обеспечить клиенту доступ к его данным.

В связи с переводом часов в разных странах и городах мира, время обновления в некоторых дата-центрах может сдвинуться на один час вперёд. Для подготовки к непредвиденным ситуациям руководство компании хочет провести учения, в ходе которых будет выбрано некоторое непустое подмножество дата-центров, и время обновления каждого из них будет сдвинуто на один час позже внутри суток (то есть, если  $u_j = h-1$ , то новым часом обновления будет 0, иначе новым часом обновления станет  $u_j + 1$ ). При этом учения не должны нарушать гарантии доступности, то есть, после смены графика обновления должно по-прежнему выполняться условие, что данные любого клиента доступны хотя бы в одном экземпляре в любой час.

Учения — полезное мероприятие, но трудоёмкое и затратное, поэтому руководство компании обратилось к вам за помощью в определении минимального по размеру непустого подходящего подмножества дата-центров, чтобы провести учения только на этом подмножестве.

### Формат входных данных

В первой строке находятся три целых числа  $n$ ,  $m$  и  $h$  ( $2 \leq n \leq 100\,000$ ,  $1 \leq m \leq 100\,000$ ,  $2 \leq h \leq 100\,000$ ) — число дата-центров компании, число клиентов компании и количество часов в сутках.

Во второй строке вам даны  $n$  чисел  $u_1, u_2, \dots, u_n$  ( $0 \leq u_j < h$ ),  $j$ -е из которых задаёт номер часа, в который происходит плановое обновление программного обеспечения на компьютерах дата-центра  $j$ .

Далее в  $m$  строках находятся пары чисел  $c_{i,1}$  и  $c_{i,2}$  ( $1 \leq c_{i,1}, c_{i,2} \leq n$ ,  $c_{i,1} \neq c_{i,2}$ ), задающие номера дата-центров, на которых находятся данные клиента  $i$ .

Гарантируется, что при заданном расписании обновлений в дата-центрах любому клиенту в любой момент доступна хотя бы одна копия его данных.



## Формат выходных данных

В первой строке выведите минимальное количество дата-центров  $k$  ( $1 \leq k \leq n$ ), которые должны затронуть учения, чтобы не потерять гарантию доступности. Во второй строке выведите  $k$  различных целых чисел — номера кластеров  $x_1, x_2, \dots, x_k$  ( $1 \leq x_i \leq n$ ), на которых в рамках учений обновления станут проводиться на час позже. Номера кластеров можно выводить в любом порядке.

Если возможных ответов несколько, разрешается вывести любой из них. Гарантируется, что хотя бы один ответ, удовлетворяющий условиям задачи, существует.

## Примеры

стандартный ввод	стандартный вывод
3 3 5 4 4 0 1 3 3 2 3 1	1 3
4 5 4 2 1 0 3 4 3 3 2 1 2 1 4 1 3	4 1 2 3 4

## Замечание

Рассмотрим первый тест из условия. Приведённый ответ является единственным способом провести учения, затронув только один дата-центр. В таком сценарии третий сервер начинает обновляться в первый час дня, и никакие два сервера, хранящие данные одного и того же пользователя, не обновляются в один и тот же час.

С другой стороны, например, сдвинуть только время обновления первого сервера на один час вперёд нельзя — в таком случае данные пользователей 1 и 3 будут недоступны в течение нулевого часа.

## Задача Н. Путешествие по музеям

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

В стране  $N$  есть  $n$  городов, соединённых  $m$  односторонними дорогами. Эта, казалось бы, непримечательная страна всё же интересна двумя своими особенностями. Во-первых, неделя здесь длится  $d$  дней. Во-вторых, в каждом городе страны  $N$  расположен ровно один музей.

Турфирма «Открытые музеи» разрабатывает новую программу для путешественников, интересующихся музеями. Работникам турфирмы известно, в какие дни недели каждый из музеев открыт для посещения. Путешествие должно начинаться в столице — городе с номером 1, причём день начала путешествия должен быть первым днём недели. Днём турист будет находиться в городе и смотреть экспозицию музея (в случае если музей, конечно, работает сегодня), а в конце дня путешествие либо заканчивается, либо турист отправляется в другой город, соединённый с текущим дорогой. Дорожная система  $N$  устроена так, что перемещение по одной дороге всегда занимает одну ночь, кроме того все дороги в стране **односторонние**. Во время путешествия разрешается посещать один город несколько раз.

Вам требуется разработать такой маршрут для путешествия, что количество **различных** музеев, которые можно посетить за его время, было бы как можно больше.

### Формат входных данных

Первая строка содержит три целых числа  $n$ ,  $m$  и  $d$  ( $1 \leq n \leq 100\,000$ ,  $0 \leq m \leq 100\,000$ ,  $1 \leq d \leq 50$ ) — количество городов, количество дорог и количество дней в неделе.

Каждая из следующих  $m$  строк содержит два целых числа  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ), обозначающие **одностороннюю** дорогу из города  $u_i$  в город  $v_i$ .

Следующие  $n$  строк содержат график работы музеев. График работы музея, находящегося в  $i$ -м городе, описывается в  $i$ -й из этих строк. Каждая строка состоит ровно из  $d$  символов «0» или «1»,  $j$ -й символ строки равен «1», если музей открыт в  $j$ -й день недели, и «0» в противном случае.

Гарантируется, для каждой пары городов  $(u, v)$  существует не более одной дороги, ведущей из  $u$  в  $v$ .

### Формат выходных данных

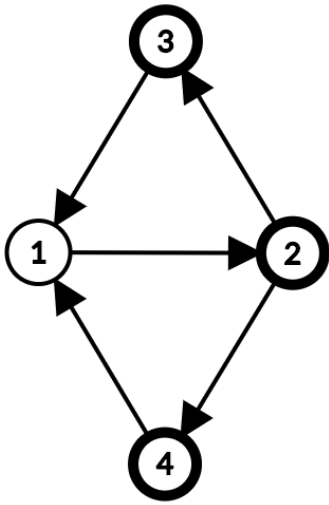
Выведите одно целое число — максимальное количество различных музеев, которые можно посетить, начав путешествие в первом городе в первый день недели.

## Примеры

стандартный ввод	стандартный вывод
4 5 3 3 1 1 2 2 4 4 1 2 3 011 110 111 001	3
3 3 7 1 2 1 3 2 3 1111111 0000000 0111111	2

## Замечание

### Пояснение к первому примеру



Максимально возможное количество музеев для посещения равно 3. Можно посетить 3 различных музея, например, так, как описано ниже.

**День 1.** Сейчас день недели 1, и турист находится в городе 1. Музей там закрыт. Ночью турист направляется в город 2.

**День 2.** Сейчас день недели 2, и турист находится в городе 2. Музей там открыт, турист его посещает. Ночью турист направляется в город 4.

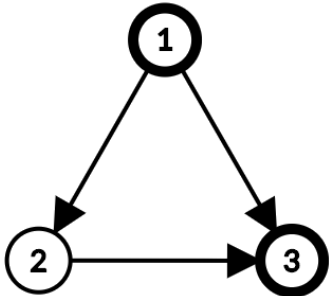
**День 3.** Сейчас день недели 3, и турист находится в городе 4. Музей там открыт, турист его посещает. Ночью турист направляется в город 1.

**День 4.** Сейчас день недели 1, и турист находится в городе 1. Музей там закрыт. Ночью турист направляется в город 2.

**День 5.** Сейчас день недели 2, и турист находится в городе 2. Музей там открыт, но турист в нём уже был. Ночью турист направляется в город 3.

**День 6.** Сейчас день недели 3, и турист находится в городе 3. Музей там открыт, турист его посещает. На этом путешествие заканчивается.

### Пояснение ко второму примеру



Максимально возможное количество музеев для посещения равно 2. Можно посетить 2 различных музея, например, так, как описано ниже.

**День 1.** Сейчас день недели 1, и турист находится в городе 1. Музей там открыт, турист его посещает. Ночью направляемся в город 2.

**День 2.** Сейчас день недели 2, и турист находится в городе 2. Музей там закрыт. Ночью направляемся в город 3.

**День 3.** Сейчас день недели 3, и турист находится в городе 3. Музей там открыт, турист его посещает. На этом путешествие заканчивается.

## Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Tinkoff Generation 2020-2021. А. Графы #1  
Водный стадион, 30 января 2020

Группа	Баллы	Дополнительные ограничения			Необх. группы	Комментарий
		$n$	$d$	График работы		
0	0	–	–	–	–	Тесты из условия.
1	9	$n \leq 10$	$d \leq 10$	–	0	
2	12	$n \leq 100$	$d \leq 10$	–	0, 1	
3	13	–	–	Каждый музей <b>работает</b> ровно один день в неделе	–	
4	17	–	–	Каждый музей <b>не работает</b> ровно один день в неделе	–	
5	13	–	$d = 2$	–	–	
6	36	–	–	–	0, 1, 2, 3, 4, 5	<b>Offline-проверка</b>

## Задача I. Экскурсия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Группа из  $n$  человек решила поехать на экскурсию. В процессе экскурсии можно заехать в некоторые из  $m$  городов.

Экскурсовод попросила каждого человека высказать свои пожелания по поводу посещения городов. Каждый человек может про какие-то города заявить, что он хочет их непременно посетить, а про какие-то — что точно не хочет.

Группа всегда путешествует вместе. Если группа заезжает в город, то все люди, заявившие, что точно не хотят его посетить, расстраиваются. Если группа не заезжает в город, то расстраиваются все люди, которые заявили, что хотят его непременно посетить.

Экскурсовод понимает, что удовлетворить все пожелания не всегда можно. Она хочет составить план, чтобы каждый человек расстроился не более одного раза.

Помогите экскурсоводу справиться с этой нелегкой задачей и составьте план посещения городов или выясните, что это невозможно.

### Формат входных данных

В первой строке входных данных содержатся три целых числа  $n$ ,  $m$ ,  $k$  — количество человек, количество городов и количество пожеланий ( $1 \leq n, m, k \leq 100\,000$ ).

В каждой из последующих  $k$  строк содержатся по два целых числа  $a$  и  $b$ , обозначающих пожелания ( $1 \leq a \leq n, 1 \leq |b| \leq m$ ). Если  $b > 0$ , то человек под номером  $a$  хочет посетить город под номером  $b$ . Если же  $b < 0$ , то человек под номером  $a$  не хочет посетить город под номером  $-b$ . Каждое пожелание указано во вводе не более одного раза, ни для кого из участников нет одновременно пожелания посетить и не посещать один и тот же город.

### Формат выходных данных

Если решения не существует, то выведите  $-1$ .

Иначе, в первой строке выходных данных выведите единственное целое число  $k$  — количество городов, которые войдут в план.

Во второй строке выведите  $k$  целых чисел — номера городов, которые следует посетить. Номера городов можно выводить в любом порядке.

Если существует несколько возможных ответов, можно вывести любой из них. Обратите внимание, что не требуется искать максимальное или минимальное  $k$ , можно вывести любой корректный ответ.

### Примеры

стандартный ввод	стандартный вывод
3 5 6 1 2 1 3 1 -4 2 3 2 4 2 5	3 2 3 5
3 3 6 1 -1 1 2 2 -2 2 3 3 -3 3 1	0

## Задача J. Интересная экскурсия

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Во Флатландии  $n$  городов, соединенных  $m$  односторонними дорогами.

Туристическая компания планирует разработать живописный циклический маршрут по дорогам Флатландии. Маршрут должен начинаться и заканчиваться в одном и том же городе и проходить по дорогам в их направлении, посещая промежуточные города. Маршрут может посещать один город несколько раз, но должен проходить по каждой дороге не более одного раза.

Каждая дорога характеризуется типом своего пейзажа, который задается числом от 1 до  $m$ . Чтобы туристический маршрут был живописным, любые две соседние дороги в этом маршруте должны иметь разный тип пейзажа. Это же требование относится к первой и последней дороге маршрута, чтобы можно было начинать путешествовать, начиная с любого города маршрута.

Помогите компании разработать удовлетворяющий этим критериям маршрут, либо выясните, что сделать это невозможно.

### Формат входных данных

Входные данные состоят из нескольких тестов. В первой строке находится число  $T$  — количество тестов ( $1 \leq T \leq 10^5$ ).

Первая строка описания каждого теста содержит два целых числа  $n$  и  $m$  — количество городов и дорог ( $2 \leq n, m \leq 2 \cdot 10^5$ ). Следующие  $m$  строк содержат по три целых числа и описывают дороги в формате  $u_i v_i c_i$  — город, из которого выходит  $i$ -я дорога, город, в который она ведет, и тип её пейзажа ( $1 \leq u_i, v_i \leq n$ ;  $1 \leq c_i \leq m$ ;  $u_i \neq v_i$ ).

Сумма  $n$  по всем тестам не превосходит  $2 \cdot 10^5$ . Сумма  $m$  по всем тестам не превосходит  $2 \cdot 10^5$ .

### Формат выходных данных

Выведите ответ для каждого теста.

Если искомого маршрута не существует, следует вывести число «-1». Иначе, выведите число  $k$  — длину маршрута ( $2 \leq k \leq m$ ). В следующей строке выведите  $k$  чисел  $e_1, e_2, \dots, e_k$  — номера дорог в том порядке, в каком они идут в этом маршруте. Все номера  $e_i$  должны быть различны. Если подходящих маршрутов несколько, можно вывести любой из них.

## Пример

стандартный ввод	стандартный вывод
3	4
5 8	3 5 6 2
1 4 1	-1
2 4 1	2
4 5 2	2 3
3 2 2	
5 3 1	
3 2 3	
5 2 2	
2 1 3	
4 5	
1 2 2	
2 3 1	
2 4 4	
4 1 2	
3 1 2	
2 3	
1 2 1	
1 2 2	
2 1 1	



## Задача К. Проблема падишаха

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Мудрый падишах внимательно следит за благополучием своих подданных, когда вершит их судьбы. В частности, на нем все заботы о вступающих в брачный возраст юношах и девушках его страны. И, как положено серьезному правителю, все по науке — перед тем, как творить молодые семьи, падишах провел Глобальное тестирование и по 100-балльной шкале определил совместимость всех юношей и девушек в совместном браке.

А дальше что? Падишах наслышан про задачу о назначении, но ему не нравится ее установка. Действительно, может ли быть спокойна его душа даже в случае всеобщего благополучия, если кому-то из подданных плохо? И можно ли жертвовать интересами хотя бы одной семьи во благо общества? Конечно, нет!

Падишаху милее другая мысль. Он хочет создать максимальное число семей, причем сделать это таким образом, чтобы минимальная совместимость в семье была максимальной. А решить эту неклассическую задачу он просит вас. Помогите падишаху!

### Формат входных данных

В первой строке входных данных содержатся два целых числа  $n$  и  $m$  — количество юношей и количество девушек соответственно ( $1 \leq n, m \leq 200$ ). Последующие  $n$  строк содержат по  $m$  целых чисел от 0 до  $10^9$  — коэффициент совместимости соответствующей пары (меньшее значение менее способствует супружеской жизни).

### Формат выходных данных

В единственную строку выходного файла выведите наименьший искомый балл, при котором возможно создание максимально возможного количества семейных пар.

### Пример

стандартный ввод	стандартный вывод
3 4 77 88 31 67 96 30 2 68 35 39 76 45	76

## Задача L. Антенна

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Для связи с Землёй членам экспедиции на Марс необходимо собрать антенну. Антенна в разобранном состоянии представляет собой  $n$  фрагментов,  $i$ -й фрагмент представляет собой штангу длиной  $s_i$  сантиметров, на которой закреплены  $m_i$  перекладин. Каждый фрагмент содержит хотя бы одну перекладину.

У каждой штанги есть начало, в котором расположен штекер, и конец, в котором расположено гнездо. Любые две штанги можно последовательно соединить, присоединив начало одной к концу другой. Для каждой перекладины известно расстояние от начала её штанги в сантиметрах. Для  $i$ -го фрагмента это расстояние может быть от 0 до  $s_i$ , значение 0 означает, что перекладина находится непосредственно в начале штанги, значение  $s_i$  — что она находится непосредственно в конце штанги. Толщиной перекладин и размерами штекера и гнезда следует пренебречь.

Чтобы корректно собрать антенну, необходимо соединить в некотором порядке все  $n$  фрагментов, при этом расстояние между любыми двумя соседними перекладинами должно быть одинаковым.

К сожалению, члены экспедиции забыли инструкцию по сборке антенны на Земле, а передать её на Марс не представляется возможным — ведь антенна ещё не собрана. Помогите исследователям!

Требуется определить, в каком порядке необходимо соединить фрагменты антенны, чтобы установить связь с Землей.

### Формат входных данных

В первой строке дано одно число  $n$  — количество фрагментов ( $1 \leq n \leq 100\,000$ ).

Далее дано описание  $n$  фрагментов. В первой строке описания фрагмента даны два целых числа  $m_i$  и  $s_i$  — количество перекладин и длина штанги в  $i$ -м фрагменте ( $1 \leq m_i \leq 100\,000$ ,  $0 \leq s_i \leq 10^9$ ). В следующей строке даны  $m_i$  целых чисел  $p_{i,j}$  — позиции перекладин,  $p_{i,j}$  равно расстоянию в сантиметрах от начала штанги до  $j$ -й перекладины на ней ( $0 \leq p_{i,1} < p_{i,2} < \dots < p_{i,m_i} \leq s_i$ ).

Сумма всех  $m_i$  не превышает 100 000.

### Формат выходных данных

Если собрать антенну указанным образом возможно, в первой строке выведите «Yes», а во второй строке выведите перестановку чисел от 1 до  $n$  — номера фрагментов в порядке, в котором их следует соединить, начало каждого следующего фрагмента в этом порядке присоединяется к концу предыдущего фрагмента. Если существует несколько подходящих ответов, можно вывести любой из них.

Если собрать антенну невозможно, в единственной строке выведите «No».

## Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Информация о проверке
1	8	$n \leq 8, m_i = 1, s_i \leq 100$		первая ошибка
2	8	$n \leq 8, s_i \leq 100$	1	первая ошибка
3	21	$n \leq 1000$	1, 2	первая ошибка
4	21	$\sum m_i > n$		первая ошибка
5	21	$s_i \leq 100$	1, 2	первая ошибка
6	21	нет	1-5	первая ошибка

## Примеры

стандартный ввод	стандартный вывод
3 1 7 3 1 8 6 2 8 1 6	Yes 2 1 3
1 1 7 5	Yes 1
1 3 10 2 5 9	No
3 1 5 3 1 3 3 1 6 3	No
4 1 5 0 1 0 0 1 3 3 1 0 0	Yes 3 2 4 1

## Задача М. Футболки на олимпиаду

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Наконец-то пришло время для финала Закрытой олимпиады школьников по программированию! Все отборы пройдены, сувенирные футболки лежат в пакетах. Вот только распределение участников по площадкам еще не спланировано.

У жюри олимпиады есть  $m$  площадок проведения. На  $i$ -й площадке лежит пакет, в котором находится  $a_i$  футболок. Это значит, что на эту площадку можно зарегистрировать не более  $a_i$  участников. Всего же среди финалистов будет  $n$  человек.

Финал Закрытой олимпиады школьников по программированию проходит в городе-герое Кленинграде. Как любой другой город Флатландии, он расположен в декартовой системе координат. Жюри знает адрес каждого финалиста — координаты  $X_i, Y_i$ , а также адреса площадок — координаты  $x_i, y_i$ .

Жюри хочет так распределить участников по точкам проведения, чтобы всем участникам хватило футболок, при этом максимальное из расстояний, которое придется пройти участнику до точки проведения, было как можно меньше. Поскольку жюри олимпиады сейчас везет на точки проведения другую сувенирку, Вас попросили провести распределение участников по площадкам.

### Формат входных данных

В первой строке заданы два целых числа  $n, m$  ( $1 \leq n, m \leq 500$ ) — количество участников олимпиады и точек проведения.

В следующих  $n$  строках заданы пары целых чисел  $x_i, y_i$  ( $1 \leq x_i, y_i \leq 10^6$ ) — координаты  $i$ -го участника олимпиады.

В следующих  $m$  строках заданы тройки целых положительных чисел  $X_i, Y_i, a_i$  ( $1 \leq X_i, Y_i \leq 10^6, 1 \leq a_i \leq n$ ) — координаты  $i$ -й площадки и ее вместимость.

Гарантируется, что сумма  $a_i$  не меньше  $n$ , а так же что она не превосходит 1000.

### Формат выходных данных

Выведите одно число — максимальное расстояние, которое придется пройти участнику до точки проведения олимпиады при оптимальном распределении. Ответ будет считаться верным, если его абсолютная или относительная ошибка не будет превосходить  $10^{-6}$ . Формально, пусть ваш ответ равен  $a$ , а ответ жюри —  $b$ . Ваш ответ считается правильным, если  $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$ .

### Примеры

стандартный ввод	стандартный вывод
3 2 1 1 2 3 3 2 1 1 1 2 2 2	1.0000000000
3 2 100 100 101 101 102 102 101 101 2 105 105 2	4.2426406871

### Замечание

В первом примере надо отправить первого участника на первую площадку, а оставшихся двух — на вторую. Тогда первый участник пройдет расстояние 0, а второй и третий — по 1.

Во втором примере надо первого и второго участников отправить на площадку 1, а третьего — на площадку 2. Тогда первый участник пройдёт расстояние 1.41421356237, второй — 0, а третий — 4.2426406871.

## Система оценки

Тесты к этой задаче состоят из девяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения			Необх. группы	Комментарий
		$n$	$m$	$a_i$		
0	0	–	–	–	–	Тесты из условия.
1	10	–	–	$a_i = n$	–	
2	11	–	$m \leq 2$	–	0	
3	15	–	$m \leq 3$	–	0, 2	
4	14	$n \leq 10$	$m \leq 10$	–	0	
5	13	$n \leq 50$	$m \leq 50$	–	0, 4	
6	7	$n \leq 200$	$m \leq 200$	–	0, 4, 5	
7	9	$n \leq 300$	$m \leq 300$	–	0, 4, 5, 6	
8	10	$n \leq 400$	$m \leq 400$	–	0, 4, 5, 6, 7	<b>Offline-проверка.</b>
9	11	–	–	–	0–8	<b>Offline-проверка.</b>

## Задача N. 2-SAT

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Формулировка 2-SAT: нужно подобрать значения  $n$  булевых переменных так, чтобы все  $m$  утверждений вида  $x_{i_1} = e_1 \vee x_{i_2} = e_2$  обратились в истину. В данной задаче вам гарантируется, что решение существует.

### Формат входных данных

Входной файл состоит из одного или нескольких тестов.

Каждый тест описывается следующим образом. На первой строке число переменных  $n$  и число утверждений  $m$ . Каждая из следующих  $m$  строк содержит числа  $i_1, e_1, i_2, e_2$ , задает утверждение  $x_{i_1} = e_1 \vee x_{i_2} = e_2$  ( $0 \leq i_j < n$ ,  $0 \leq e_j \leq 1$ ). Ограничения: сумма всех  $n$  не больше 100 000, сумма всех  $m$  не больше 300 000.

### Формат выходных данных

Для каждого теста выведите строку из  $n$  нулей и единиц — значения переменных. Если у данной задачи 2-SAT есть несколько решений, выведите любое.

### Пример

стандартный ввод	стандартный вывод
1 0	0
2 2	01
0 0 1 0	000
0 1 1 1	
3 4	
0 1 1 0	
0 0 2 1	
1 1 2 0	
0 0 0 1	

## Задача О. Лабиринт

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В одном из уровней компьютерной игры вы попали в лабиринт, состоящий из  $n$  строк, каждая из которых содержит  $m$  клеток. Каждая клетка либо свободна, либо занята препятствием. Стартовая клетка находится в строке  $r$  и столбце  $c$ . За один шаг вы можете переместиться на одну клетку вверх, влево, вниз или вправо, если она не занята препятствием. Вы не можете перемещаться за границы лабиринта.

К сожалению, ваша клавиатура крайне близка к поломке, поэтому вы можете переместиться влево не более  $x$  раз и вправо не более  $y$  раз. При этом ограничений на перемещения вверх и вниз нет, поскольку клавиши, используемые для движения вверх и вниз, всё ещё в идеальном состоянии.

Теперь вы для каждой клетки поля решили установить, можно ли выбрать такую последовательность нажатий, которая приведёт вас из стартовой в эту клетку. Посчитайте, сколько клеток поля обладают таким свойством.

### Формат входных данных

Первая строка содержит два целых числа  $n, m$  ( $1 \leq n, m \leq 2000$ ) — количество строк и столбцов в лабиринте, соответственно.

Вторая строка содержит два целых числа  $r, c$  ( $1 \leq r \leq n, 1 \leq c \leq m$ ) — номер строки и столбца, на пересечении которых расположена стартовая клетка.

Третья строка содержит два целых числа  $x, y$  ( $0 \leq x, y \leq 10^9$ ) — максимальное количество перемещений влево и вправо, соответственно.

Следующие  $n$  строк содержат описание лабиринта. Каждая из этих строк имеет длину  $m$  и состоит только из символов '.' и '\*'. В  $i$ -й строке  $j$ -й символ соответствует клетке лабиринта с номерами строки и столбца  $i$  и  $j$ , соответственно. Символ '.' соответствует свободной клетке лабиринта, а символ '\*' — клетке с препятствием.

Гарантируется, что стартовая клетка не занята препятствием.

### Формат выходных данных

Выведите одно число — количество клеток лабиринта, достижимых из стартовой, включая её саму.

### Пример

стандартный ввод	стандартный вывод
4 5 3 2 1 2 ..... .***. ...** *.....	10

## Задача Р. Электрическая схема

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Паша — начинающий техник, но уже поставил себе большую цель собрать собственный компьютер. Первая непростая задача — научиться собирать электрическую схему.

Схема, которую собрал Паша, состоит из несколько проводов. Каждый провод — это отрезок, который соединяет две точки на плоскости с целыми координатами, лежащими в диапазоне  $[1, 10^9]$ .

В схеме есть провода двух цветов:

- красные провода: эти провода должны иметь вид горизонтального отрезка, то есть если провод соединяет две точки  $(x_1, y_1)$  и  $(x_2, y_2)$ , то выполнено, что  $y_1 = y_2$ ;
- синие провода: эти провода должны иметь вид вертикального отрезка, то есть если провод соединяет две точки  $(x_1, y_1)$  и  $(x_2, y_2)$ , то выполнено, что  $x_1 = x_2$ .

Обратите внимание, что если провод соединяет две одинаковые точки, то он может быть как красным, так и синим. Также в Пашиной схеме никакие два провода одного цвета не могут пересекаться, то есть любые два отрезка проводов одного цвета не могут содержать общих точек.

Недоработка Пашиной схемы состоит в том, что его провода не были изолированы, и поэтому в точках пересечения проводов разных цветов возникли искры, которые Паша увидел. Он записал все точки, в которых он увидел искру. У него получилось множество из  $n$  различных точек. После чего он разобрал схему и пошёл спать.

Утром, когда Паша увидел на листочке множество из  $n$  точек, в которых он увидел искру, ему стало интересно, сколько проводов он использовал, собрав эту схему. К сожалению, он ничего не запомнил, поэтому он решил узнать, какое минимальное количество проводов он мог использовать в своей схеме. Помогите ему узнать это число, а также расположить эти провода так, чтобы в получившейся схеме искры возникли в тех же самых точках.

### Формат входных данных

В первой строке находится одно целое число  $n$  ( $1 \leq n \leq 1000$ ) — количество точек, в которых Паша увидел искру.

В следующих  $n$  строках находится по два целых числа  $x$  и  $y$  ( $1 \leq x, y \leq 10^9$ ) — координаты очередной точки. Гарантируется, что все точки различны.

### Формат выходных данных

Выведите описание электрической схемы в следующем формате:

Сначала выведите  $h$  — количество горизонтальных красных проводов ( $0 \leq h$ ). В следующих  $h$  строках выведите по 4 целых числа  $x_1, y_1, x_2, y_2$  — координаты двух точек  $(x_1, y_1)$  и  $(x_2, y_2)$ , которые соединяет очередной красный провод. Поскольку отрезки горизонтальные, должно быть выполнено  $y_1 = y_2$ . Также должно быть выполнено  $1 \leq x_1, y_1, x_2, y_2 \leq 10^9$ .

Потом выведите  $v$  — количество вертикальных синих проводов ( $0 \leq v$ ). В следующих  $v$  строках выведите по 4 целых числа  $x_1, y_1, x_2, y_2$  — координаты двух точек  $(x_1, y_1)$  и  $(x_2, y_2)$ , которые соединяет синий очередной провод. Поскольку отрезки вертикальные, должно быть выполнено  $x_1 = x_2$ . Также должно быть выполнено  $1 \leq x_1, y_1, x_2, y_2 \leq 10^9$ .

Никакие два отрезка одного цвета не должны иметь общих точек. Множество точек, в которых Паша мог увидеть искру, если бы он построил такую схему, должно совпадать с данным во входных данных множеством точек.

Количество отрезков ( $h + v$ ) должно быть минимально возможным. Можно легко показать, что ответ всегда существует. Если существует несколько возможных ответов, выведите любой.



## Примеры

стандартный ввод	стандартный вывод
4 2 2 2 4 4 2 4 4	2 5 2 1 2 1 4 5 4 2 2 1 2 5 4 5 4 1
4 2 1 3 2 2 3 1 2	4 2 1 2 1 3 2 3 2 1 2 1 2 2 3 2 3 3 1 2 1 2 3 2 3 2 2 3 2 1

## Замечание

В первом примере Паша мог собрать такую схему:

В этой схеме по 2 провода каждого цвета: красные из (5, 2) в (1, 2) и из (1, 4) в (5, 4), синие из (2, 1) в (2, 5) и из (4, 5) в (4, 1). Заметим, что он увидит искры ровно в тех точках, которые он записал (обозначены желтым цветом на картинке). Например, искру в точке (2, 4) он увидит, так как в этой точке пересекаются второй красный провод и первый синий. Можно доказать, что нужно не меньше 4-х проводов, чтобы получить схему, нужную Паше.

## Задача Q. Магнитные подушки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Город будущего застроен небоскребами, для передвижения между которыми и парковки транспорта многие тройки небоскребов соединены треугольной подушкой из однополярных магнитов. Каждая подушка соединяет ровно 3 небоскреба и вид сверху на нее представляет собой треугольник, с вершинами в небоскребах. Это позволяет беспрепятственно передвигаться между соответствующими небоскребами. Подушки можно делать на разных уровнях, поэтому один небоскреб может быть соединен различными подушками с парами других, причем два небоскреба могут соединять несколько подушек (как с разными третьими небоскребами, так и с одинаковым). Например, возможны две подушки на разных уровнях между небоскребами 1, 2 и 3, и, кроме того, магнитная подушка между 1, 2, 5.

Система магнитных подушек организована так, что с их помощью можно добраться от одного небоскреба, до любого другого в этом городе (с одной подушки на другую можно перемещаться внутри небоскреба), но поддержание каждой из них требует больших затрат энергии.

Требуется написать программу, которая определит, какие из магнитных подушек нельзя удалять из подушечной системы города, так как удаление даже только этой подушки может привести к тому, что найдутся небоскребы из которых теперь нельзя добраться до некоторых других небоскребов, и жителям станет очень грустно.

### Формат входных данных

В первой строке входного файла находятся числа  $N$  и  $M$  — количество небоскребов в городе и количество работающих магнитных подушек соответственно ( $3 \leq N \leq 100000$ ,  $1 \leq M \leq 100000$ ). В каждой из следующих  $M$  строк через пробел записаны три числа — номера небоскребов, соединенных подушкой. Небоскребы пронумерованы от 1 до  $N$ . Гарантируется, что имеющиеся магнитные подушки позволяют перемещаться от одного небоскреба до любого другого.

### Формат выходных данных

Выведите в выходной файл сначала количество тех магнитных подушек, отключение которых невозможно без нарушения сообщения в городе, а потом их номера. Нумерация должна соответствовать тому порядку, в котором подушки перечислены во входном файле. Нумерация начинается с единицы.

### Примеры

стандартный ввод	стандартный вывод
3 1 1 2 3	1 1
3 2 1 2 3 3 2 1	0
5 4 1 2 3 2 4 3 1 2 4 3 5 1	1 4